Name

Ayush Choubey

Location

Bhubaneswar, India (GMT + 5:30)

Education

3rd Year Undergraduate, KIIT University, Bhubaneswar,India

Email / IRC / Jabber

E-Mail- ayush.choubey@gmail.com

IRC: ayusun

WWW / Blog

theayusun.blogspot.com

Synopsis

None of the project is perfect. Each of them strives to become better, so as to provide its user more features and become more usable. Over the period of time they will require some new features some big, some small but essential which will make the software more user-friendly more productive and more secured

This GSoC project aims for the same thing, to provide more power to user and make PMA more user-friendly, more secured and more productive. This project contains many smaller modules and features that have to be Implemented in phpMyadmin and every module has got its own use.

Benefits to the users

The project will bring various benefits to the Users. Some features are aimed to make PMA more User Friendly while some are aimed to give user more power and features and hence upgrade the capabilities of PMA. This Project also aims at making PMA more stable which is again beneficial for user. And the overall outcome will be that users will be able to manage their mysql part in much more efficient manner.

Project Details

Following are the details of the project that i will be doing with some sample snippets or the plan that i've thought for making this project successful:

- **Find and replace by column** : As the name suggest we should be able to find and replace the data in a particular column. As far as interface is concerned, i think a dropdown menu, containg list of all column names of the table, a Textbox to find the text, a another textbox to keep replaced text and a checkbox to give an option of case sensitivity, a Find button (to find all the matching text) and a replace button (to make all the replacements) will be sufficient.

  As far as code is concerned, i've thought of making another private function getFindAndReplace() which will return the string containing html for the interface. To get all the column names we can use the function getcolumns(defined in PMA_Table class). GetFindAndReplace() will be called from getTableHeader(well this is the place i've thought of,will make necessary changes, if required) and thus including them along with table

headers. The find button of this interface will send a POST request to sql.php which will return list of the condition_array(the one which used by the checkbox using PMA_Util::getUniquecondition) . This condition_arrays can be used as the selector in the jquery and all the checkboxes which can match them will be checked. Now the user has the choice to select all the necessary changes that he wants and then click on replace button and all the selected changes will be replaced at ones

- **On user creation, warn if the user already exists :**
  As quoted earlier in the tracker
  “If you create an user with the username of an already exsiting account, but with a different 'host' setup, PMA will create the user 'bravely', which is correct.
  But this may sometimes be an unwanted thing”.

  In this case we cannot use mysql.user table directly, because in some cases a user may not have permission to directly use mysql.user table. So we can store these user data in phpmyadmin database and every time before creating the user we can have a check that if that user is present in pma's table or not ; by using PMA_DBI_try_query() function.

  According to the output of the query we can either give an error message or create a user(and subsequently do an entry in PMA database)

- **Hide/unhide tables** : When lots of results are produced by a software, then there is always a probability to select a wrong one. Even though PMA has very good GUI that minimises this problem, but still there is always a probablity. To further minimize it, it is the best to discard the result that are not required, but we can't actually discard any table can we! So let's hide all those tables that are not actually required or seldom they are used.

  First, the interface part, Since it is one of Action for the table, So its better to keep it grouped in the Action Column of the table. Upon clicking which a Jquery event will occur and the entire row get hidden. Now, the next thing is to remember this result i.e which row has been hidden, so that the next time the user visits it's database, he finds the same setting. For that, I would like to use phpMyAdmin DataBase(PMA__table_uiprefs) and add a new field into it (visibility status) for that purpose, and on hiding for the first time a new Record will be created. So whenever we fetch all the tables, First all the tables will be created will be displayed on the page and then all the table which should be hidden will get hidden automatically(By the use of jquery, ofcourse). **And to unhide the tables, a button will be used in the top left corner of the table which will use “.show()” by using proper selectors.**
- **reCAPTCHA support on login panel** : We all know how advantageous recaptchas are. Recaptcha's client part will be made available in auth() function, while its validation part will be done in authcheck function. Recaptcha public and private key must be set using the configuration setup script and they will be stored in config.inc.php file(obviously in encoded format).

- **Ajax dialog for editing a view** : As far as view is concerned, they should just be used to view data and editing should be minimized to insertion only(however if required, will add operation of updation and deletion also,in case of simple views; but that shouldn't be done). This module will deal with using Ajax Dialog bix to do insertion(and other operations if required). Currently, we insert data into view using same tbl_change.php that we use for table. In this module, i will be Ajaxifying the things by bringing up a dialog box using Ajax, however no major change in source code will occur. The new form will still submit the credentials to the tbl_replace.php and every other working will remain same and thus code duplicity will be reduced by using the same code and functionality wise the PMA will improve
- **Export view as if it was a table** : In the current implementation, exporting view means using the same sql command that was used to create it. Like

  CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `qwerty` AS select `testbin`.`id` AS `id`,`testbin`.`bin` AS `bin` from `testbin`;

  This actually is a bit incorrect as it still depend on the origional table. Hence in this module i will be giving Views the same treatment that we give to Tables and hence in the end we would be getting first create statement followed by the insert statement.
- **Improve "Relational View" Interface** : In the current implementation there is just on single <select> element that displays all the possible choices(based on index created). The current model works perfect when the number of tables are less, but when the number of tables increases, then select element becomes quite cluttered and then it becomes difficult to select on of them.

  In the new model, instead on one, we will use two select element, one will be for selecting a table and other will be for selecting a column. This will surely reduce the cluttering and give a better user experience.

  From the code point of view, Every information is stored in $selectboxall array, In tbl_relation.php. So first we have to break it into individual fields and store it into array. Later on after posting data we need to merge them and every other functionality will remain same.
- **Single Row Editor**: **Value column too small** : This will again deal with user experience, Probably changing it into a text-area which should be resized according to text,so that no scrollbar appears.

  A sample code example will be : http://jsbin.com/adebol which have to be integrated in the present code
- **Preview SQL instead of executing it** : It is one of the feature that **DB2** control centre also contains and i was also looking for a feature like it in phpMyAdmin also. This module is bit big, because functionalities are are scattered over different pages, but that's what makes it less troublesome. From interface point of view I will be adding a “showSQL” button besides every Go button and from the fuctionality point of view, every fuction will receive a parameter “showsql” which if true will not execute PMA_DBI_try_query (well mostly what i've seen) and thus in the end only the sql query will appear. In a nutshell, this is the most straight forward approach i can think of, probably a bit of planning will be

required with the mentor, but overall a solution with least possible changes

- **Allow SQL query textarea growable / resizable** : For Sql Query we use two diffeent types of text area, one is code mirror plugin and other is normal textarea.
  I believe for the best user exerience it's better if the text area grows automatically according to the text in it which will eliminate the use of scroll bars.

  Regarding textarea :http://jsbin.com/adebol : probably one of the solution, however there are many other

  Regarding Codemirror : http://codemirror.net/demo/resize.html : A valid example was given in code mirror website and after a few tinkering with the code this module should be easily finished.

- **Configurable menus** : As it was suggested
  “The phpMyAdmin configuration storage could hold Group definitions that describe which menu items are available to each group, and which users are part of which group (with a default group).”
  On the same ground work, we have to make necessary changes in Menu.class.php to pass the group id or (may be storing it into $cfg variable and then using it into Menu.class.php) and a few modification has to made as what to keep and what not to keep according to group.

Deliverables

The final output will be the implementation of all these features, with minimum code duplication and without changing much of the existing codebase. The code will use PEAR standards and will be checked using “phpcs”. And upon completion of every individual module, they will be tested with every possible test cases, so as to remove any bug that remains with the implementation.

Project Schedule

I have taken sufficient margin for the modules and believe that they should be more than enough and i couldn't say about it, but if possible i would like to implement few other features as well, However the above 11 are the ones i was most interested in and i have divided them in the manner according to the time and depending upon the previous modules that i would have completed.

28 May – 16 May	Get to know Mentors, Read Documentation, Compile the list of functions that would be used by me in various modules, Strategise the coding approaches, Implement “**Single Row editor**” and “**Allow SQL query textarea growable / resizable**” as they are among the smallest modules and are almost readymade

	available. Have it reviewed by the Mentor.
17 June – 30 June	Coding period starts. Will start implementing Modules in the order **Recaptcha support for login panel** **Improve Relational View Interface**
1 July – 10 July	**Export View as if it were a table** **Find and Replace by Column**
10-July – 25 July	**Preview SQL instead of executing it**
25 July – 29 July	Testing of features implemented and have them Review by the Mentor, Implementation of the codes on the basis of feedback given by the mentor
29 july-1 Aug	Midterm Evaluation
1 Aug- 10 Aug	**Ajax Dialog for Editing a View** **Hide/unhide tables**
11 Aug- 11 Sept	**On user creation** **warn if the user already exists** **Configurable menus.**
16 Sept-23 Sept	Final Testing, improve Documentation, Final Review By Mentors
24 Sept	Submission for final Evaluation

Time

My Exams will be over on 18 May and then my summer vacation will start which will be of about 2 months for which i haven't made any plans and thus can spend more than 45 hours/ week at ease

After that my classes will start and we have to do some minor projects, but since i've already prepared one in my 6th semester, i won't have to prepare one again and thus at that time also i can spend about 40 hours a week.

Motivation

After High School, when i was admitted into college, that was probably the first time i ever used a linux distro and from that point i don't know how many open source software were used by me; and every time, they amazed me. And now after three years, i think its time that i pay back to the FOSS world with what i am good at, and GSoC is probably the best and grandest way for any student to enter into the FOSS world.

Bio

I am Ayush Choubey, 20 year old, 3rd year Computer Science undergraduate at KIIT University,

Bhubaneswar. I just love open source software(partly because they are free :D) and i use them alot(probably everytime). I like programming, specially tweaking with the codes so as to understand them more clearly. Apart from programming and computer related stuffs i like Basketball, cricket, animes and chatting.

Other than this i am the organiser of GDG Bhubaneswar and thus organise various competations and events related to google technologies and Opensource. I am also one of the heads of the college IT society Konnexions and there also i do more or less the same thing in addittion to taking classes.

Experiences

- I have been developing web pages using HTML and javascript since class 10th and have started developing web application since 1st year using php-Mysql. I started off with small things like match-making and simple login based webpages and then started advancing.
- After that i contributed in my friends website(techahoy.com) which used php, mysql and JS. Planning to work on it again after GSoc
- Created a project named “The Basilisk-HR Operation Manager” for the IBM- The Great Mind Challenge. It was inspired by Orange HRM. The Code used J2EE, DB2, Jquery
- Created a chatterbot named Grin as my class 12 final project. The project was inspired by Eliza. The project was written in VB6.0 and used MS- ACCESS
- Currently while contributing for phpmyadmin, i learnt using GIT and other softwares like phpcs while contributing for it.

Open source contributions

For last three years, i've been using linux as my main Operating system and i am totally passionate about them, but never got the chance to contribute towards it (in the form of coding), however in college IT society and in GDG meetups, i always encourage students to use Open Source Software and that is how i contribute to Open Source in my own little way.

phpMyAdmin contributions

I started off a bit late becuase of my university exams and practicals and then took quite a bit of time in getting the hang of git and phpMyAdmin code base, however now i am quite comfortable with both of them.
My few Contributions are :

- https://github.com/phpmyadmin/phpmyadmin/pull/265
- https://github.com/phpmyadmin/phpmyadmin/pull/268

The above two were not merged because of better solutions that were available

- https://github.com/phpmyadmin/phpmyadmin/pull/275
  which was updated in https://github.com/phpmyadmin/phpmyadmin/pull/286

- https://github.com/phpmyadmin/phpmyadmin/pull/283
- https://github.com/phpmyadmin/phpmyadmin/pull/285

The above 3 have been proposed are being reviewed

Favorite phpMyAdmin feature

My favourite phpMyadmin feature has got be the export and import feature. I don't know, my how much time it has saved, while developing website for techahoy i remember how big databases were and if it hadn't been for this particular feature, i don't know what would have happened to the websites

phpMyAdmin improvement

Currently phpMyAdmin deals mostly with DataBase part using php. But if it can become a complete project management system, then it would be even better. One particular feature that after learning git came to my mind was- “what if we can clone projects using phpMyadmin and if it can push and pull into the REPO automatically!!”. Well the project is definately big and requires a lot of planning but definately would love to work on this one.